Project #4:
Machine Learning Loan Predictor

Group 4: Brent Beachtel, Justin Bein, Tico Brown,
Jim Haugen, Celina Kamler, and Nataliia Shevchenko

Professor Kevin Lee

Project Overview

Main Goal
Create a webpage that will predict whether a loan will be approved, based on user-entered data, using Machine Learning (ML).

Secondary Goal
Within the webpage, display a visual representation of comparative information so applicants can see how their parameters fare against all other loan applicants

Data Source:

- A csv sourced from Kaggle with 614 observations pulled from mortgage applications

Audience:

- Banks, loan officers, and prospective loan applicants interested in increasing their chances of approval

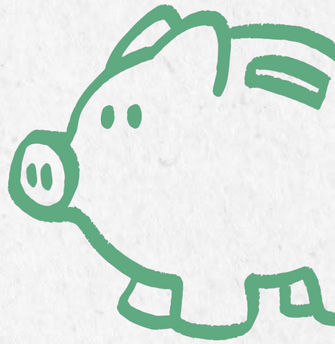
Tools/ Modules used:
Pandas, sklearn, numpy, Tableau, Flask, & Tensorflow

Why Use ML?

Historically, lending decisions in the US have been heavily biased in favor of white males.

Despite the passage of the Fair Housing Act (1968) and the Equal Credit Opportunity Act (1974), discriminatory practices persist. We hypothesize that machine learning could circumvent human bias in lending practices.

A caveat: Because models are be trained on data derived from previous, biased loan underwriting, the models themselves can inherit that bias.



Our Process

ETL

Clean the data, impute missing continuous values, and assign correct data types

EDA

Explore the shape of the data using pandas, numpy and Tableau visualizations

ML Tests

Test out a different Machine Learning model and different optimizers to find the most accurate one

Deploy

Deploy the model to our website using html, css, and Flask

ETL

Our Extract, Transform, and Load (ETL) process:

- Review the csv and look for missing values
- Impute missing continuous values using K Nearest Neighbors
- Drop missing categorical values
- Update data types
- Look for outliers
- Export clean data file for the team to use

EDA

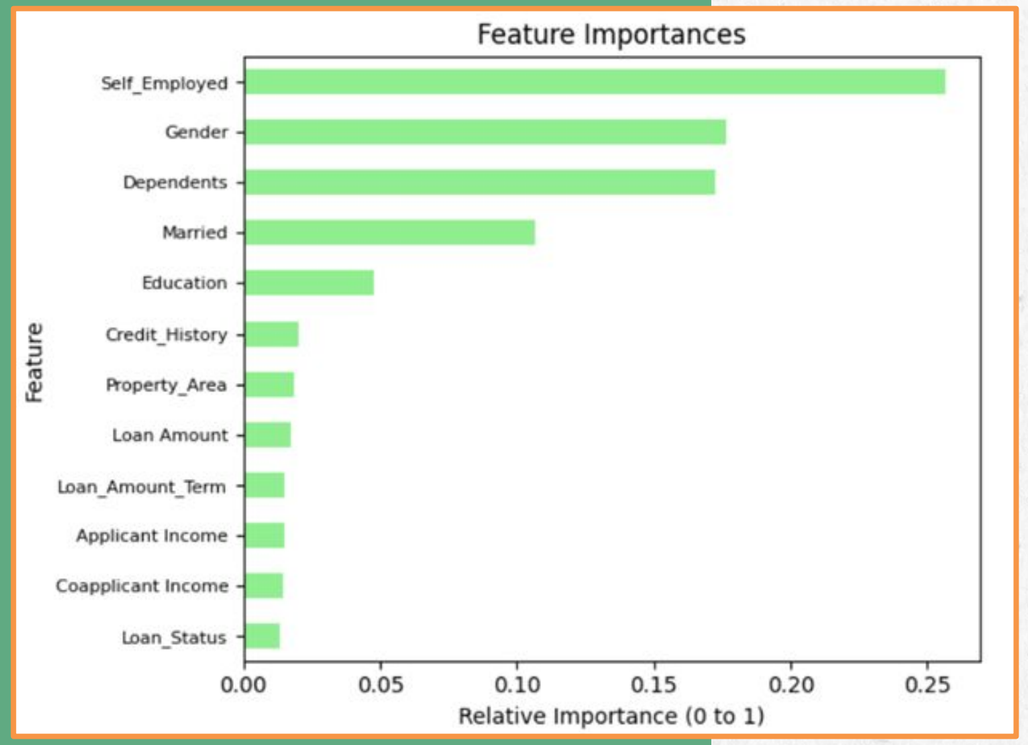
Exploratory data analysis was conducted using Python and Tableau.

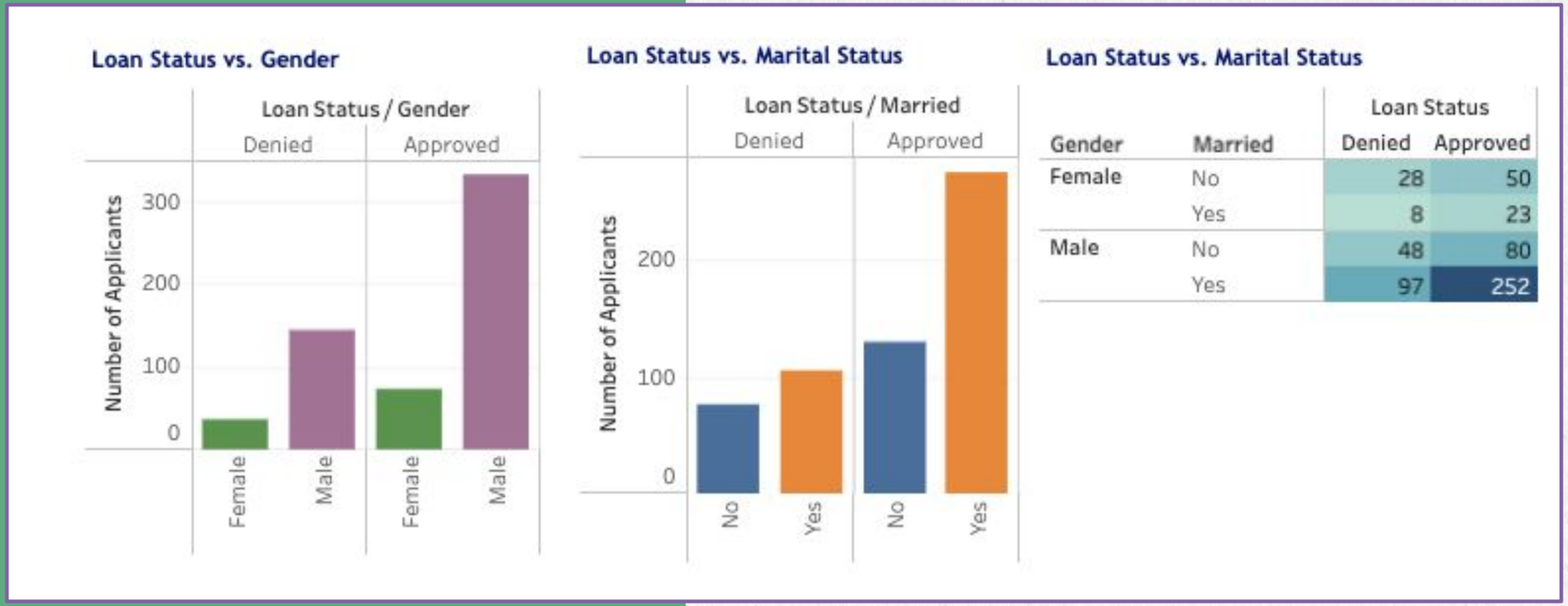
Python:

- Examined descriptive statistics
- Determined relative importances of each feature

Tableau:

- Created visuals of demographic features
- Created visuals of loan characteristics
- Created a loan predictor





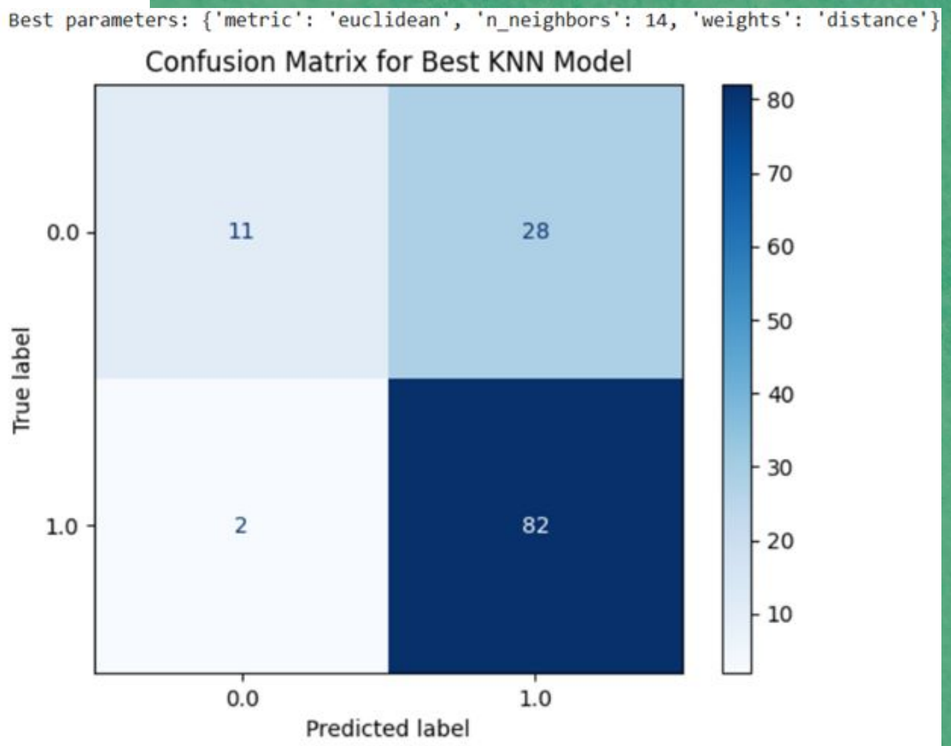
Loan Status vs. Marital Status

Gender	Married	Loan Status: Denied	Loan Status: Approved
Female	No	28	50
	Yes	8	23
Male	No	48	80
	Yes	97	252

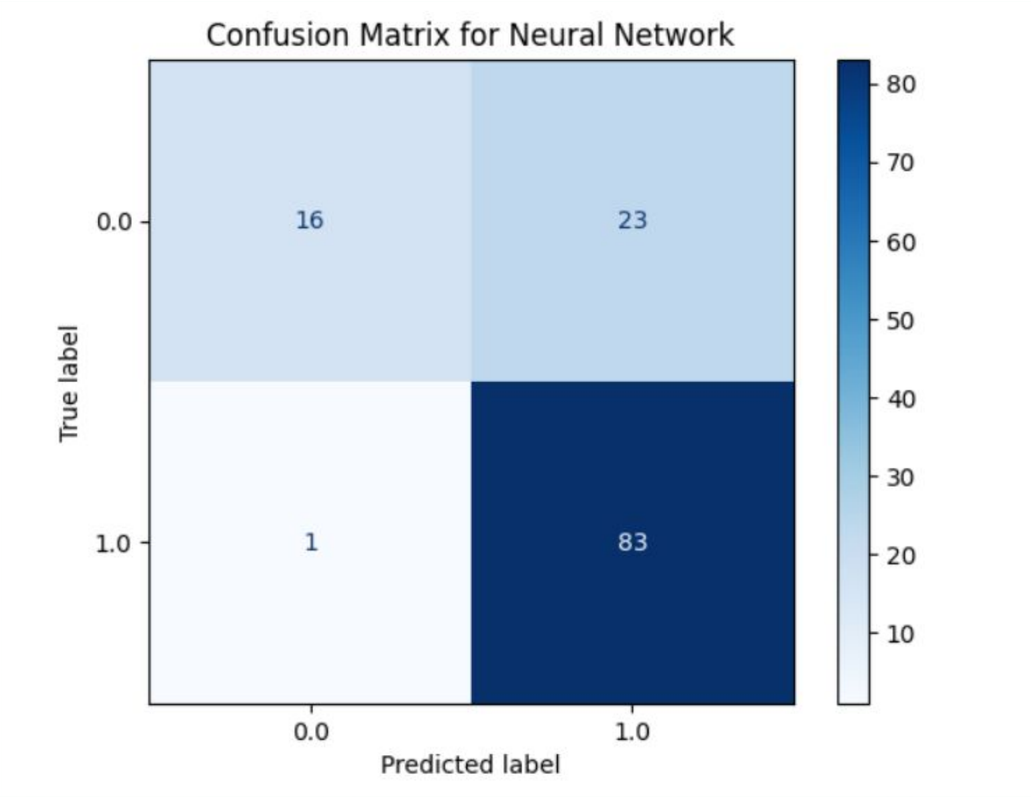
Machine Learning Models

Model	Accuracy
Logistic Regression	80%
Neural Networks	82%
K Nearest Neighbors	76%
Gradient Boosting Model	78%
Random Forest	78%

Best parameters: {'metric': 'euclidean', 'n_neighbors': 14, 'weights': 'distance'}



	precision	recall	f1-score	support
0.0	0.85	0.28	0.42	39
1.0	0.75	0.98	0.85	84
accuracy			0.76	123
macro avg	0.80	0.63	0.63	123
weighted avg	0.78	0.76	0.71	123



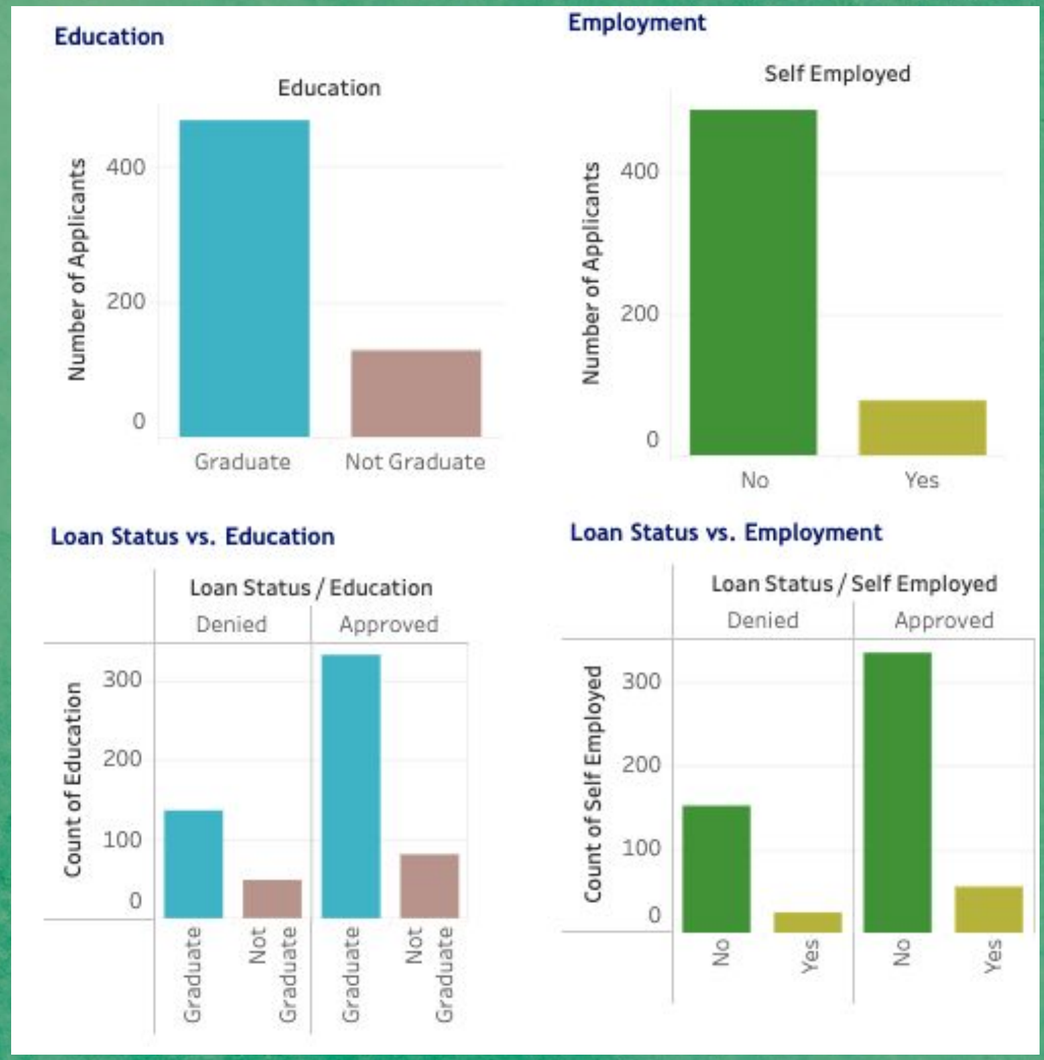
80% accuracy

Trust your credit to Neural Models!

 # Model Architecture

Layer (type)	Output Shape	Param #
dense_14 (Dense)	(None, 256)	5376
batch_normalization_4 (BatchNormalization)	(None, 256)	1024
dropout_10 (Dropout)	(None, 256)	0
dense_15 (Dense)	(None, 128)	32896
batch_normalization_5 (BatchNormalization)	(None, 128)	512
dropout_11 (Dropout)	(None, 128)	0
dense_16 (Dense)	(None, 64)	8256
batch_normalization_6 (BatchNormalization)	(None, 64)	256
dropout_12 (Dropout)	(None, 64)	0
dense_17 (Dense)	(None, 32)	2080
batch_normalization_7 (BatchNormalization)	(None, 32)	128
dropout_13 (Dropout)	(None, 32)	0
dense_18 (Dense)	(None, 1)	33

Tableau Visuals



Linear Regression

Correlation Heatmap Analysis

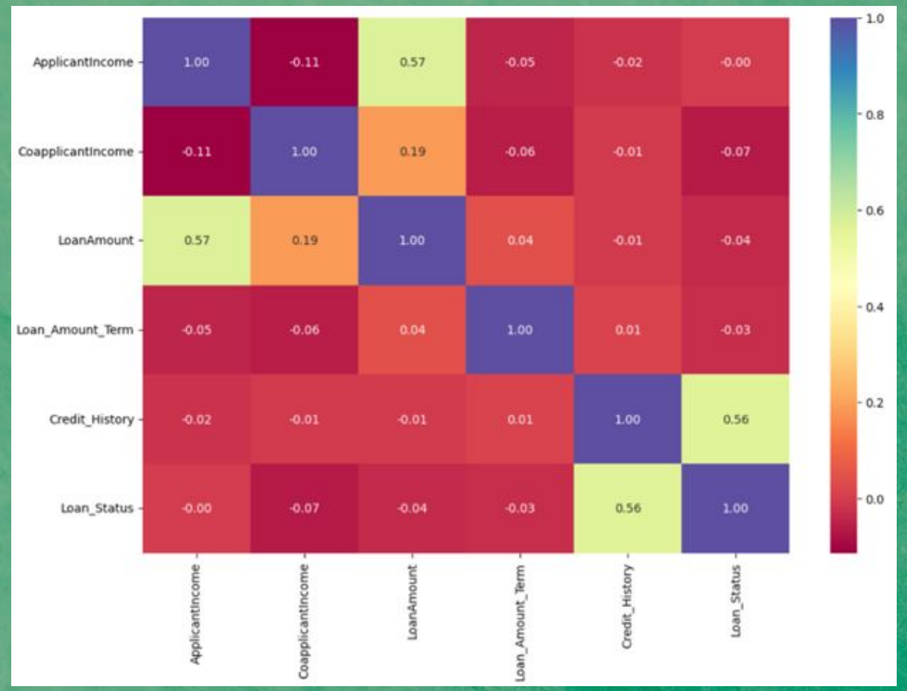
Cleaned data set (599 rows)

- Applicant Income v. Loan Amount
- Co-applicant Income v. Loan Amount

Histogram & Boxplot Analysis

- Performed on both
- Both right skewed & outliers
- Dropped Incomes > $30k

Result: Second data set without outliers created (583 rows)



Linear Regression Models

- Fitted four models using the two data sets
- Examined r^2 & RMSE for each model
- Best regression formula:
- Maximum Loan Amount = 103.57 +

  0.007927 * Applicant Income

Deployment

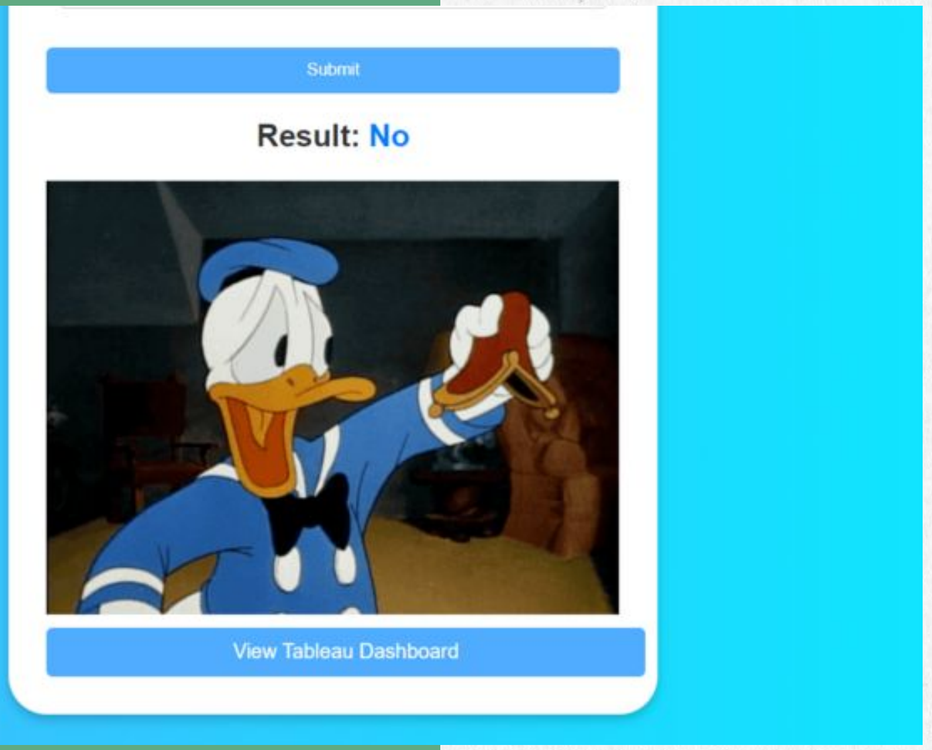
- Initialize Flask application
- Load model.pkl, scaler.pkl, and choices.pkl
- Create df with correct shape to feed model (this included setting nan values for non user prompted inputs)
- model.predict for a binary classification on button click in html

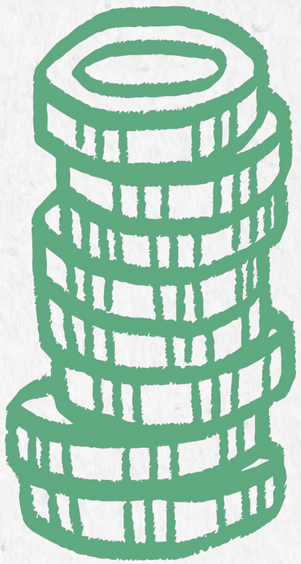
```
def predict(input_data):
    """
    Helper function to make predictions based on input data.
    """
    # Create a DataFrame with the correct columns
    X = pd.DataFrame(columns=choices.keys())

    # Map input data to the correct columns
    X.loc[0, 'ApplicantIncome'] = input_data.get('Combined_income', np.nan)
    X.loc[0, 'CoapplicantIncome'] = 0  # Set CoapplicantIncome to 0 explicitly
    X.loc[0, 'LoanAmount'] = input_data.get('Requested_amount', np.nan)
    X.loc[0, 'Loan_Amount_Term'] = 360  # Example value, adjust as necessary

    # Handle categorical variables
    X.loc[0, 'Credit_History'] = 1 if input_data['Credit_history'] == 'Yes' else 0
    X.loc[0, 'Gender_Female'] = 1 if input_data.get('Gender') == 'Female' else 0
    X.loc[0, 'Gender_Male'] = 1 if input_data.get('Gender') == 'Male' else 0
```

Website Demo





Summary & Next Steps

- Income (high), self-employment status (N), & prior credit history (Y) are all strong predictors of loan approval
- Married males with prior credit history who are not self-employed are more likely to get approved for a loan
- We still have room for growth in eliminating bias from the loan approval process!

Next steps: gather more recent data from different sources (credit unions, community banks, etc) to see if bias can be further reduced